

CLAIMS

What is claimed is:

1. An apparatus for graphics processing in a handheld device, the apparatus comprising:

at least one of: a transform engine and a vertex shader, such that the at least one of the transform engine and the vertex shader is capable of receiving a plurality of vertex information and generating a plurality of vertices, wherein each of the vertices has a corresponding bin identifier;

a plurality of view frame factors defining a clipping region, upon which the vertex shader compares each of the plurality of vertices, wherein when one of the vertices is within the clipping region, a clip identifier is generated for that vertex using the corresponding bin identifier;

the vertex shader operably coupleable to a clipping module on a central processing unit (CPU) wherein when one of the vertices is within the clipping region, the vertex shader provides the vertices having a corresponding clip identifier to the clipping module which thereupon generates supplemental vertices and the vertex shader receives the supplemental vertices therefrom; and

a vertex buffer operably coupled to the vertex shader wherein the vertex buffer receives the plurality of vertices and the supplemental vertices from the clipping module on the CPU and stores the plurality of vertices therein.

2. The apparatus of claim 1 further comprising:

a setup engine operably coupled to the vertex shader wherein the setup engine receives the plurality of vertices and the supplemental vertices.

3. The apparatus of claim 2 further comprising:
a rasterization engine coupled to the setup engine for receiving a setup engine output; and
a pixel operation module coupled to the rasterization engine and the frame buffer such
that the pixel operation module transforms a rasterization engine output signal and
provides a viewable output display signal to the frame buffer.
4. The apparatus of claim 3 further comprising:
a display device operably coupled to the frame buffer such that upon generation of a
viewable display from the viewable output display signal, the viewable display
may be provided to the display device from the frame buffer.
5. The apparatus of claim 4 wherein the display device is a liquid crystal display.
6. The apparatus of claim 1 wherein the plurality of view frame factors include at
least one of the following: a view frustum region and a guard band region.
7. The apparatus of claim 1 further comprising:
a clip counter having a clip counter value such that the clip counter value is incremented
each time one of the vertices is deemed within the clipping region and once all of
the plurality of vertices have been generated by the vertex shader, the clip counter
value is provided to a central processor such that the central processor may
properly retrieve the corresponding number of clip identifiers.
8. The apparatus of claim 1 wherein the clipping module is disposed on a central
processor and the apparatus is coupled to the central processor across a bus and the transform
engine is disposed within the vertex shader.

9. A method for graphics processing in a handheld device, the method comprising:
receiving a plurality of vertex information;
performing a vertex transform on the vertex information to generate a plurality of
vertices, wherein each of the vertices has a corresponding bin identifier and a
corresponding vertex identifier;
comparing each of the vertices with a plurality of view frame factors wherein the view
frame factors define a clipping region;
generating a clip identifier for each of the vertices within the clipping region using the
corresponding bin identifier and the corresponding vertex identifier;
providing the clip identifiers to a clipping module; and
receiving a plurality of supplemental vertices generated from the plurality of vertices
within the clipping region.
10. The method of claim 9 further comprising:
incrementing a clip counter value each time one of the vertices is deemed within the
clipping region; and
providing the clip counter value to a central processor such that the central processing
knows the number of clip identifiers to be provided to the clipping module.
11. The method of claim 9 further comprising:
performing a viewport transform on the plurality of vertices and the plurality of
supplemental vertices.
12. The method of claim 11 further comprising:
providing an output signal from the viewport transform to a setup engine.
13. The method of claim 12 further comprising:

generating a setup engine output; and
providing the setup engine output to a rasterization engine.

14. The method of claim 13 further comprising:

generating a rasterization engine output signal;
providing the rasterization engine output signal to a pixel operation module;
transforming the rasterization engine output signal to generate a viewable output display
signal; and
providing the viewable output display signal to a frame buffer.

15. The method of claim 14 further comprising:

providing the viewable display signal from the frame buffer to a display device.

16. The method of claim 11 wherein prior to the step of performing the viewport transform, the method further comprises:

receiving a list of triangles to be rendered, wherein the triangle is defined by three
vertices;

for each of the triangles in the list of triangles, determining if each of the vertices
defining the triangle is within the clipping region; and

discarding the vertices within the clipping region thereby generating a list of only
triangles having vertices within the clipping region.

17. The method of claim 16 further comprising:

prior to performing the viewport transform, defining a plurality of rendering regions and
selecting one of the plurality of rendering regions such that only the vertices
within the rendering regions are provided to the viewport transformer;

generating one of a plurality of viewable display signals representing a viewable portion of the rendering region; and
writing the viewable display signal to a frame buffer, wherein the viewable display signal is directed to the selected rendering region.

18. The method of claim 17 further comprising:

performing the viewport transform for each of the plurality of rendering regions;
generating the plurality of viewable display signals representing the plurality of viewable portions of the rendering region;
writing the plurality of viewable display signals into the frame buffer; and
providing a complete viewable output display signal from the frame buffer to a display device, wherein the complete viewable display output signal includes the plurality of viewable display signals, thereby capable of generating a complete output display.

19. The method of claim 18 further comprising:

utilizing a z buffer having a storage capacity of one of the plurality of rendering regions to generate the plurality of viewable display signals.

20. A method for graphics processing in a handheld device comprising:
receiving a plurality of vertices;
transforming the plurality of vertices to generate a corresponding bin identifier and a
vertex identifier for each of the plurality of vertices;
for each vertex, determine if the vertex is within a clipping region;
if the vertex is within a clipping region:
incrementing a clip counter; and
generating a clip identifier using the corresponding bin identifier and the vertex
identifier;
clipping the vertices having the clip identifier to generate a plurality of supplemental
vertices having a corresponding bin identifier based on each of the vertices having
a clip identifier;
updating a vertex list to include all vertices not within the clipping region and the
supplemental vertices; and
performing a viewport transformation on the vertex list.

21. The method of claim 20 further comprising:
dividing a full frame into a plurality of frame portions;
defining the clipping region to include all vertices outside one of the plurality of frame
portions such that the vertex list only includes vertices within the one of plurality
of the frame portions;
after performing the viewport transformation on the vertex list, generating a viewable
output display signal representing the frame portion; and
writing the viewable output display signal in a frame buffer.

22. The method of claim 21 further comprising:

for each of the plurality of frame portions:

defining the clipping region to include all vertices outside one of the plurality of frame portions such that the vertex list only includes vertices within the one of the plurality of frame portions;

after performing the viewport transformation on the vertex list, generating a viewable output display signal representing the frame portion; and

writing the viewable output display signal in a frame buffer; and

providing the plurality of viewable output display signal from the frame buffer to an output display.

23. The method of claim 22 further comprising:

during the generation of viewable output display signal, utilizing a z buffer having a storage capacity of one of the plurality of rendering regions to generate the plurality of viewable display signals.

24. The method of claim 20 wherein the step clipping the vertices is performed by a central processing unit.

25. The method of claim 20 wherein the step of transforming the plurality of vertices and the step of viewport transformation is performed on a graphics processor.

26. A graphics processing device for a handheld device comprising:
- a vertex shader capable of receiving a plurality of vertex information and generating a plurality of vertices, wherein each of the vertices has a corresponding bin identifier;
 - a central processing unit coupled to a vertex shader;
 - a plurality of view frame factors defining a clipping region, upon which the vertex shader compares each of the plurality of vertices, wherein when one of the vertices is within the clipping region, a clip identifier is generated for that vertex using the corresponding bin identifier;
 - a clip counter having a clip counter value such that the clip counter value is incremented each time a vertex is deemed within the clipping region and once all of the plurality of vertices have been generated by the vertex shader, the clip counter value is provided to the central processor such that the central processor may properly retrieve the corresponding number of clip identifiers;
 - the vertex shader provides the vertices having a corresponding clip identifier to the central processor executing a clipping module such that the clipping module generates a plurality of supplemental vertices and the vertex shader receives the supplemental vertices from the central processor; and
 - a vertex buffer operably coupled to the vertex shader wherein the vertex buffer receives the plurality of vertices and the supplemental vertices from the vertex shader and stores the plurality of vertices therein.
27. The graphics processing device of claim 26 further comprising:

a setup engine operably coupled to the vertex shader wherein the setup engine receives the plurality of vertices and the supplement vertices;

a rasterization engine coupled to the setup engine for receiving a setup engine output; and

a pixel operation module coupled to the rasterization engine and the frame buffer such that the pixel operation module transforms a rasterization engine output signal and provides a viewable output display signal to the frame buffer.

28. The graphics processing device of claim 27 further comprising:

a display operably coupled to the frame buffer such that upon generation of a viewable display from the viewable output display signal, the viewable display may be provided to the display from the frame buffer.

29. The graphics processing device of claim 26 wherein the plurality of view frame factors include at least one of the following: a depth factor, a view frustum region and a guard band region.

30. The graphics processing device of claim 26, wherein the vertex shader includes a transform engine such that the transform engine generates the plurality of vertices.